

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 09-237186

(43)Date of publication of application : 09.09.1997

(51)Int.Cl. G06F 9/38
G06F 9/32

(21)Application number : 08-045801

(71)Applicant : HITACHI LTD

(22)Date of filing : 04.03.1996

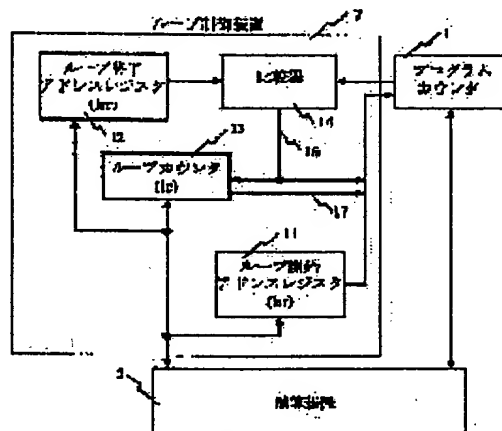
(72)Inventor : NAKADA KEIMEI
TANAKA KAZUHIKO
NOJIRI TORU
NISHIOKA KIYOKAZU
KUROKAWA NORIYOSHI
KOJIMA KEIJI

(54) LOOP CONTROLLER AND PROCESSOR

(57)Abstract:

PROBLEM TO BE SOLVED: To reduce the number of execution cycles required for the loop control of a pipeline or a processor having plural computing elements by counting the repeated frequency of plural continuous instructions and specifying and maintaining the address of a start position.

SOLUTION: A loop start address, a loop end address and repeated frequency are respectively set up in registers 11, 12 and a loop counter 13. When a comparator 14 detects coincidence between the contents of a program counter 1 and the contents of the end address register 12 and the contents of the loop counter are 'non-0', the contents of the counter 13 are reduced, the contents of the loop start address register 11 are transferred to the counter 1 to execute branching. When the contents of the counter 13 are '0', the contents of the counter 13 are not reduced and the contents of the register 11 are not transferred to the counter 1, so that the counter 1 indicates an instruction within a loop range.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

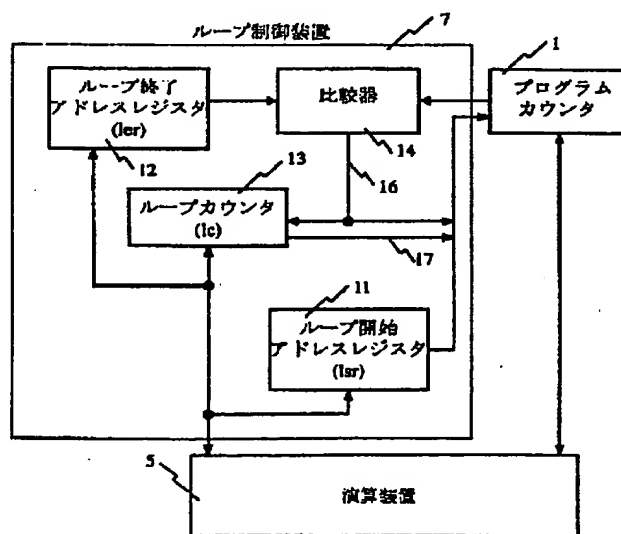
[Date of registration]

(43)公開日 平成9年(1997)9月9日

審査請求 未請求 請求項の数 3 O.L (全 7 頁)

最終頁に続く

【解決手段】演算装置５とデータの受渡しを行なう経路を持ち、命令により任意の値を設定可能なループ開始アドレスを維持するループ開始アドレスレジスタ１１と、ループ終了アドレスを維持するループ終了レジスタ１２と、反復回数をカウントするループカウンタ１３と、ループ範囲の終了を検出する比較器１４とからなる。



【特許請求の範囲】

【請求項 1】記憶装置に格納されたプログラムを構成する命令の中からある時点で処理を要する命令を特定する手段、前記命令を特定する手段により特定された命令を取り込む手段、前記取り込む手段により取り込まれた命令により指示された演算を実行する手段により構成されるプロセッサにおいて、連続した複数命令を複数回にわたり反復して実行するループ処理の初期設定手段、前記初期設定手段に結合された反復の回数をカウントする手段、前記初期設定手段に結合された前記複数命令の終了位置を維持する手段、及び前記複数命令の開始位置のアドレスを明示的に指定する前記初期設定手段に結合された前記複数命令の開始位置を維持する手段を含むことを特徴とするループ制御装置。

【請求項 2】命令の実行を複数のステージに分割して行なう手段、及び前記命令を実行する手段で 1 ステージの処理が終了し次の命令の実行に必要な準備が整い次第次の命令の実行を開始する手段を含むパイプライン構成のプロセッサで、請求項 1 のループ制御装置を持つプロセッサ。

【請求項 3】一つの処理を指示する命令を複数個連結して長命令とし、記憶装置に格納されたプログラムを構成する前記長命令の中からある時点で処理を要する前記長命令を特定する手段、前記長命令を特定する手段により特定された長命令を取り込む手段、前記取り込む手段により取り込まれた長命令を複数の命令に分割する手段、及び前記分割する手段により分割された複数の命令をそれぞれ実行する複数の演算する手段を含む VLIW (Very Long Instruction Word) プロセッサで、請求項 1 のループ制御装置を持つプロセッサ。

【発明の詳細な説明】**【0001】**

【発明の属する技術分野】本発明はループ制御装置およびプロセッサに関する。

【0002】

【従来の技術】ループは反復して実行される命令ブロックから構成され、ベクトル演算や行列演算、画像処理等に用いられる。

【0003】ソフトウェアによってループを実現するには、ループ回数を管理するためにレジスタまたはメモリを割り当て、ループの最後で割り当てたレジスタまたはメモリの値を 1 減らし、結果が非 0 であればループの先頭に分岐することにより行う。しかし、ソフトウェアのみによってループを実現する場合、ループ制御を行うためだけに減分/比較/分岐処理がループ毎に必要となり、オーバーヘッドが生じる。

【0004】このようなオーバーヘッドを削減する手法として、特開平 2-183831 号公報に開示された手法がある。この手法では、ループ初期化命令を用意し、ル

ープ初期化命令でループ反復回数及びループ範囲内に存在する命令数を指定することで、ループ範囲内からループ制御を行う上で必要な減分/比較/分岐処理に要する実行サイクル数を削減している。

【0005】ところが、この手法では必ずループ範囲の直前にループ初期化命令を配置する必要があり、命令記述位置を制限する。この制限により、パイプラインや複数の演算器を持つプロセッサにおいて処理の高速化に重要な役割を果たす命令記述位置の最適化が困難となる。

【0006】パイプラインを持つプロセッサで命令記述位置の重要性を示す例として図 1 及び図 2 に命令 101B と 101C の位置が入れ替わっている以外全く同様の処理を行なう二つのプログラムを示す。プロセッサが加算命令の実行を 2 ステージのパイプラインで処理する場合、それぞれのプログラムを実行した際のパイプラインの動作は図 3 及び図 4 に示したようになり、実行に必要なサイクル数が 2 サイクル異なる。これは命令 101B が命令 101A の結果を、命令 101D が命令 101C の結果をそれぞれ必要とする為、図 1 に示したプログラムの実行では命令 101A、101C の終了を待つ必要が生じ、何も処理を行なっていない空きスロット 111A、111B がパイプラインに発生するのに比べ、図 2 に示したプログラムの実行では命令 101A の実行中に命令 101A の結果を必要としない命令 101C を実行するため、命令 101A、101C の終了を待つのに必要な空きスロットがパイプラインに発生しないからである。

【0007】このように、パイプライン処理を行なうプロセッサでは空きスロットをできるだけ発生させないよう命令記述位置を最適化することが実行サイクル数の削減、すなわち処理の高速化に重要な役割を果たす。

【0008】

【発明が解決しようとする課題】パイプラインや複数の演算器をもつプロセッサのループ制御に要する実行サイクル数を削減することである。

【0009】

【課題を解決するための手段】本発明では、ループ制御を行うためにハードウェアを設け、そのハードウェアの初期化を行う専用のループ初期化命令を用いる。ループ制御を行うハードウェアはループ開始アドレスを維持するループ開始アドレスレジスタ、ループ終了アドレスを維持するループ終了アドレスレジスタ、残りの反復回数を管理するループカウンタ及び現在実行している命令のアドレスとループ終了アドレスの一致を検出する比較器により構成される。ループ開始アドレスレジスタ、ループ終了アドレスレジスタ、ループカウンタは、ループ初期化命令に相当するループ開始アドレス指定命令、ループ終了アドレス指定命令、反復回数指定命令によりそれぞれ内容を直接設定可能なように構成する。

【0010】まず、ループ開始アドレス指定命令、ルー

ブ終了アドレス指定命令、反復回数指定命令の3つから成るループ初期化命令によってループ制御ハードウェアの設定が行われる。ループ開始アドレス指定命令により指定されたループ開始アドレスがループ開始アドレスレジスタに設定され、ループ終了アドレス指定命令により指定されたループ終了アドレスがループ終了アドレスレジスタに設定され、さらに反復回数指定命令により指定された反復回数がループカウンタに設定される。

【0011】現在プロセッサが読み込もうとしている命令のアドレスを管理するプログラムカウンタの内容とループ終了アドレスレジスタの内容は比較器によって常時比較され、比較器が一致を検出すると、ループカウンタの内容が非0の場合には、ループカウンタの内容が減分され、さらにループ開始アドレスレジスタの内容がプログラムカウンタ転送され、分岐が実行される。一方、比較器が一致を検出し、ループカウンタの内容が0であった場合には、ループカウンタの内容は減分されず、ループ開始アドレスレジスタの内容もプログラムカウンタに転送されない為、プログラムカウンタはループ範囲の次の命令を指すこととなり、ループ処理は終了する。

【0012】ループ開始アドレス指定命令によりループ開始アドレスを直接指定する為、ループ初期化命令を必ずしもループ範囲の直前に記述する必要がなくなり、ループ初期化命令記述位置の最適化が可能となる。

【0013】これにより、課題となっていたパイプラインや複数の演算器を持つプロセッサのループ制御に要する実行サイクル数の削減が実現可能となる。

【0014】

【発明の実施の形態】実施例として本発明を用いたプロセッサの構成例を図5に示す。プロセッサが読み込もうとしている命令の存在するアドレスを管理しているプログラムカウンタ1、記憶装置3とプロセッサ内部のデータ転送を制御する記憶制御装置2、プログラムやデータを保持する記憶装置3、記憶装置3から記憶制御装置2を介して読み込んだ命令を解読して演算装置5の制御を行なう命令解読装置4、実際に演算処理を行なう演算装置5、演算に必要なデータや演算結果を保持する汎用レジスタ6、ループの制御を行なうループ制御装置7で構成されている。

【0015】プロセッサは記憶装置3に存在するプログラムにしたがって動作を行う。プロセッサはプログラムカウンタ1の示すアドレスを用いて記憶装置3から命令を読み込み、通常では命令を読み込んだ後に次の命令アドレスを指すようにプログラムカウンタ1を更新する。しかし、演算装置5またはループ制御装置7から値が代入された場合には、プログラムカウンタ1の値はこれらの装置から代入された値となり分岐が実行される。

【0016】プロセッサが命令を記憶装置3から読み込む場合、プログラムカウンタ1の値が命令アドレスとして記憶制御装置2に伝えられ、記憶制御装置2はこの命

令アドレスにしたがって記憶装置3より命令をプロセッサに読み込む。

【0017】読み込まれた命令は記憶制御装置2を介して命令解読装置4に送られ、命令解読装置4は命令にしたがって適切な制御信号を発生し、演算装置5を制御する。

【0018】演算装置5は汎用レジスタ6、ループ制御装置7及びプログラムカウンタ1とのデータのやりとりが可能であり、さらに記憶制御装置2を介することで記憶装置3とのデータのやりとりも可能であるため、命令解読装置4からの制御信号にしたがってこれらデータの演算や転送を行う。演算装置5はパイプライン処理を用いて演算処理を行い、また演算処理中に転送処理を独立して行うことが可能である。

【0019】ループ制御を行うハードウェアであるループ制御装置7の内部構成は図6に示したようになっており、ループ開始アドレスを維持するループ開始アドレスレジスタ(lsr)11、ループ終了アドレスを維持するループ終了アドレスレジスタ(ler)12、残りの反復回数を維持するループカウンタ(lc)13、及び現在プロセッサが読み込もうとしている命令のアドレスを維持するプログラムカウンタ1の内容とループ終了アドレスレジスタ12の内容の一致を検出する比較器14で構成される。

【0020】ループ制御装置7の初期化を行う為には専用の転送命令を用い、この転送命令はループ開始アドレスレジスタ11を設定するmove lsa命令、ループ終了アドレスレジスタ12を設定するmove lea命令、ループカウンタ13を設定するmove lc命令の3命令により構成される。

【0021】プログラムカウンタ1の内容とループ終了アドレスレジスタ12の内容は比較器14で常時比較されており、比較器14が出力する一致検出信号16はループカウンタ13のカウンタ動作制御及びループ開始アドレスレジスタ11からプログラムカウンタ1への値の書き込み制御に用いられる。ループカウンタ13は比較器14からの一致検出信号16により、その時点でのループカウンタ13の値が非0である場合には値を減分する。しかしその時点でのループカウンタ13の値が0である場合には値の更新は行わない。また、ループカウンタ13はカウンタ値が非0の場合非0検出信号17を出力し、非0検出信号17は比較器14の出力する一致検出信号16と共にループ開始アドレスレジスタ11からプログラムカウンタ1への値の書き込み制御に用いられる。

【0022】比較器14の出力する一致検出信号16及びループカウンタ13の出力する非0検出信号17が同時に発生するとループ開始アドレスレジスタ11の値がプログラムカウンタ15へ書き込まれ、ループ開始アドレスへの分岐が実行される。

【0023】このように構成することで、ループカウンタ13が0になるまで、ループカウンタ13を減分しながらループ開始アドレスレジスタ11の示すアドレスとループ終了アドレスレジスタ12の示すアドレスの間に存在する命令ブロックを繰り返し実行することになるので、ループ処理を実現することができる。さらに、ループ開始アドレスレジスタ11を設定する専用のmove lsa命令を用意したことで、ループ初期化命令に相当する命令をループ範囲の直前以外に記述することが可能となる。

【0024】次に、図5に示したプロセッサのプログラム実行時における命令実行タイミングの例をループ開始アドレス指定命令の存在しないループ制御方式をもつプロセッサと比較して説明する。ループ開始アドレス指定命令の存在しないループ制御方式をもつプロセッサ用のプログラム例とその実行タイミングを図7、図8に、図5に示したプロセッサのプログラム例とその実行タイミングを図9、図10に示す。ループ開始アドレス指定命令の存在しないプロセッサではループ初期化命令に相当する転送命令23、24をループ範囲の直前に記述することでループ開始アドレスを示す必要がある。この場合、instA20の実行に必要なステージがnでinstB21がinstA20の実行結果を必要とする場合、図8の命令実行タイミングが示すように、instA20の実行が終了するまでのnサイクルの間instB21以降の実行は開始されず、instA20の実行結果を必要としないループ初期化命令に相当する転送命令23、24も実行されない。

【0025】ところが本発明を用いた手法ではループ初期化命令によって明示的にループ開始アドレスを指定可能な為、instA20とinstB21の間にループ初期化命令に相当する転送命令22～24を記述でき、instA10の実行期間中にループ初期化命令に相当する転送命令22～24を実行可能となり、空きスロットを削減できるため、効率の良い実行が可能となる。この結果、実施例ではn=3の場合、1サイクル、nが4以上の場合2サイクル実行サイクルを削減することができる。

【0026】別の実施例として、本発明を用いたVLIW方式のプロセッサの構成例を図11に示す。

【0027】このプロセッサは基本的に図5に示したプロセッサと同様の動作を行う。しかし、演算装置5A～5D及びこれに関係する命令解読装置4A～4D、汎用レジスタ6A～6Dが存在し、独立した命令をそれぞれの演算装置5A～5Dに対して同時に発行可能である。これに伴い記憶装置3から記憶制御装置2を介して読み込まれる命令は、図12に示すように4個の命令フィールド9A～9Dにより構成される長命令10となっており、1命令フィールドには図5に示したプロセッサの1命令が記述される。また、プログラムカウンタ1も長命令単位でプロセッサが現在読み込もうとしているアドレ

スを管理している為、分岐も長命令単位で行われる。

【0028】記憶装置3からプロセッサに読み込まれた長命令は、まず記憶制御装置2を介して命令分配装置8に送られ、命令分配装置8で独立した4個の命令に分解された後に、それぞれ命令解読装置4A～4Dに送られる。命令解読装置4A～4Dはそれぞれ演算装置5A～5Dの制御を行うことで命令の実行を行う。

【0029】演算装置5A～5Dにはそれぞれ専用の汎用レジスタ6A～6Dが接続されている他、プログラムカウンタ1、ループ制御装置7、記憶制御装置2が接続され、さらに演算装置5A～5D間で相互に接続されている。演算装置5A～5Dは接続された装置からデータを得て演算処理をしたり接続された装置間のデータ転送をすることで、命令の実行を行う。

【0030】ループ制御装置7の動作は図5に示したプロセッサと同じであるが、ループ開始アドレスレジスタ11及びループ終了アドレスレジスタ12に設定するアドレスは長命令単位となる。

【0031】次に、ループ開始アドレス指定命令の存在しないループ制御方式を用いたVLIW方式のプロセッサでの命令スロットの使用例を図13に、図12に示したプロセッサでの命令スロットの使用例を図14に示し、両者の違いを説明する。

【0032】図13に示したように命令スロット42、44、48がnop(No Operation)となる場合でも、ループ開始アドレス指定命令の存在しないプロセッサではループ初期化命令に相当する転送命令をループ範囲の直前に記述する必要があるため、ループ初期化命令に相当する転送命令を記述する為の命令スロット53、54を別に用意する必要がある。しかし本発明を用いた方式では、ループ初期化命令をループ範囲の直前に記述する必要がない為、図14に示すように命令スロット42、44、48をループ初期化命令に相当する転送命令の為に用いることができる。この結果、長命令1命令分プログラムステップを削減でき、プログラムサイズ及び実行サイクルの削減に役立つ。

【0033】

【発明の効果】本発明により、プログラム中におけるループ初期化命令記述位置の最適化が可能となり、ループ初期化命令によるオーバーヘッドを削減できる。特にVLIW方式のプロセッサに本発明を適用した場合には、プログラム中に含まれるnop命令の命令スロットをループ初期化命令に活用でき、長命令を削減できる可能性があるため、実行サイクルの削減だけではなく、プログラムサイズの削減にも効果がある。

【図面の簡単な説明】

【図1】パイプライン構成のプロセッサを効率良く利用しない命令列の説明図。

【図2】パイプライン構成のプロセッサを効率良く利用する命令列の説明図。

【図3】図1の命令列実行時のパイプラインの動作例の説明図。

【図4】図2の命令列実行時のパイプラインの動作例の説明図。

【図5】本発明のループ制御装置を含むプロセッサのブロック図。

【図6】図5に示したプロセッサ構成例のループ制御装置部分のブロック図。

【図7】ループ開始アドレスを指定できない場合のプログラム例の説明図。

【図8】図7に示したプログラム実行時のパイプラインの動作例の説明図。

【図9】ループ開始アドレスを指定できる場合のプログラム例の説明図。

【図10】図9に示したプログラム実行時のパイプラインの動作例の説明図。

【図11】本発明のループ制御装置を含むVLIWプロセッサのブロック図。

【図12】図11に示したVLIWプロセッサで用いる長命令の説明図。

【図13】図11に示したVLIWプロセッサでループ開始アドレスを指定できない場合の命令スロット使用例の説明図。

【図14】図11に示したVLIWプロセッサでループ開始アドレスを指定できる場合の命令スロット使用例の説明図。

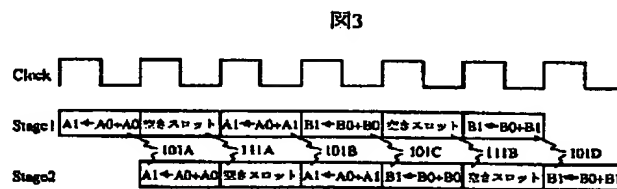
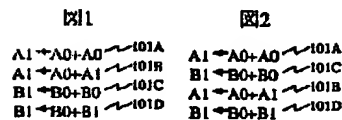
【符号の説明】

1…プログラムカウンタ、5…演算装置、7…ループ制御装置、11…ループ開始アドレスレジスタ、12…ループ終了アドレスレジスタ、13…ループカウンタ、14…比較器、16…一致検出信号、17…非0検出信号。

【図1】

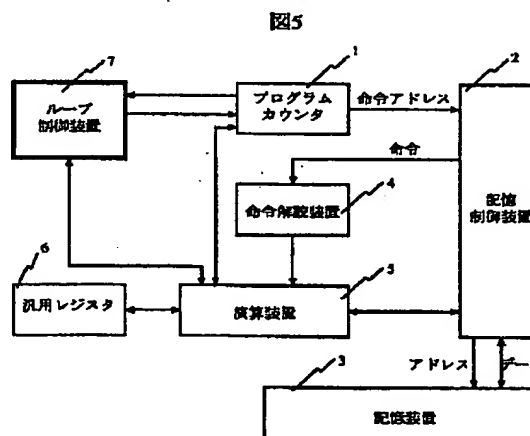
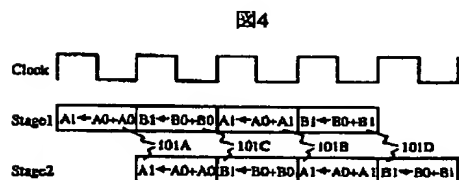
【図2】

【図3】

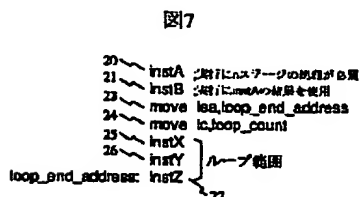


【図4】

【図5】

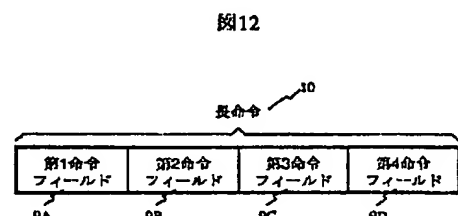
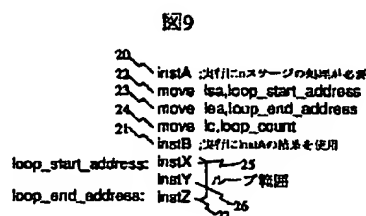


【図7】

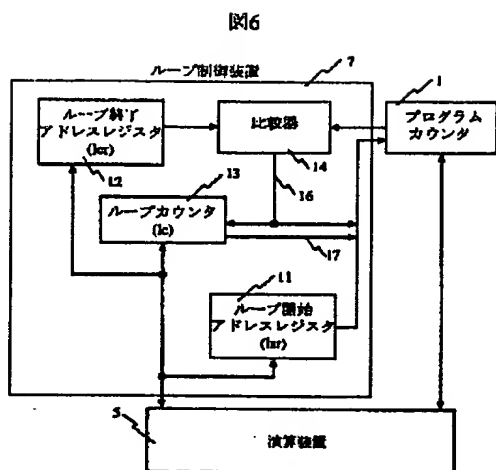


【図9】

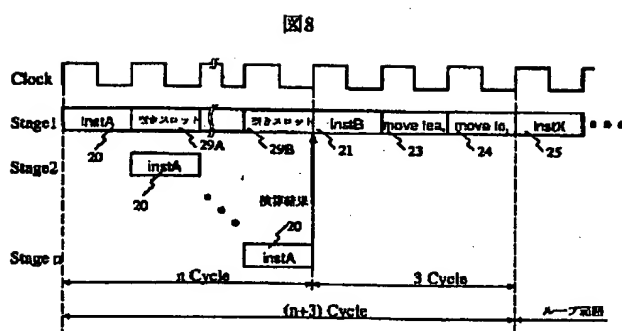
【図12】



【図6】



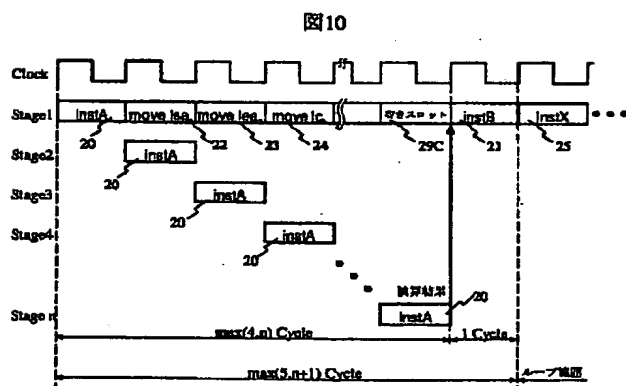
【図8】



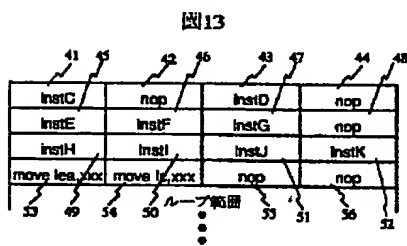
【図11】

図11

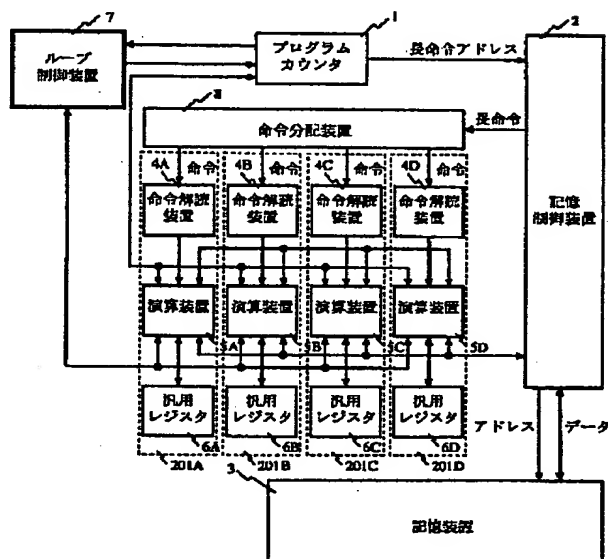
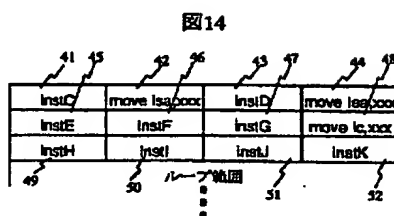
【図10】



【図13】



【図14】



フロントページの続き

(72)発明者 西岡 清和
神奈川県川崎市麻生区王禅寺1099番地株式
会社日立製作所システム開発研究所内

(72)発明者 黒川 能毅
神奈川県川崎市麻生区王禅寺1099番地株式
会社日立製作所システム開発研究所内

(72)発明者 小島 啓二
神奈川県川崎市麻生区王禅寺1099番地株式
会社日立製作所システム開発研究所内